ADHIYAMAAN COLLEGE OF ENGINEERING (AUTONOMOUS)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

TOPIC: CAR RESALE VALUE PREDICATION

Team ID-PNT2022TMID08070

TEAM LEADER :

AJAYSIVAN

TEAM MEMBERS :

GNANENDRA PRASAD
DRAVIDKUMAR
ALFREDSAMSTEPHEN
ARAVINDNANDHA

CONTENTS

3. **Ideation & ProposedSolution**

4. **Requirement Analysis**

5. **Project Design**

6. **Project Planning & Scheduling**

7. **Coding & Solutioning**

8. **Testing**

9. **Results**

    a. Project Overview
    b. Purpose

    a. Existing Problem
    b. References
    c. Problem Statement Definitions

    a. Empathy Map Canvas
    b. Ideation& Brainstorming
    c. Proposed Solution
    d. Problem Solution Fit

a. Functional Requirement
b. Non-Functional Requirement

a. Data Flow Diagrams
b. Solution& Technical Architecture
c. User Stories

a. Sprint Planning & Estimation
b. Sprint Delivery Schedule
c. Reports From JIRA

a. Feature 1
b. Feature 2

a. Testcases
b. User Acceptance Testing

9.1 Performance Metrics

10. **Advantages & Disadvantages**

11. **Conclusion**
12. **Future Scope**
13. **Appendix**

ACKNOWLEDGEMENT

On the submission of this report on "CAR RESALE VALUE PREDICTION", we would like to extend our gratitudeand sin cere thanks to our Mentor ANAJANA DEVI , Assistant Professor, Department of ELECTRONICS AND COMMUNICATION ENGINEERING(ECE)for his constant motivation and support during the course. We truly appreciate and value his good guidance and encouragement from the beginning to the end of this this project. We are indebted to his help for having helped us shape the problem and providing insights towards the solution.

1. Introduction

In this paper, we investigate the application of supervised machine learning techniques to predict the price of used cars in Mauritius. The predictions are based on historical data collected from daily newspapers. Different techniques like multiple linear regression analysis, k-nearest neighbours, naïve bayes and decision trees have been used to make the predictions. The predictions are then evaluated and compared in order to find those which provide the best performances. A seemingly easy problem turned out to be indeed very difficult to resolve with high accuracy. All the four methods provided comparable performance. In the future, we intend to use more sophisticated algorithms to

Predicting the price of used cars in both an important and interesting problem. According to data obtained from the National Transport Authority [1], the number of cars registered between 2003 and 2013 has witnessed a spectacular increase of 234%.
From 68, 524 cars registered in 2003, this number has now reached 160, 701. With difficult economic conditions, it is likely that sales of second-hand imported

(reconditioned) cars and used cars will increase. It is reported in [2] that the sales of new cars has registered a decrease of 8% in 2013.

In many developed countries, it is common to lease a car rather than buying it outright. A lease is a binding contract between a buyer and a seller (or a third party – usually a bank, insurance firm or other financial institutions) in which the buyer must pay fixed instalments for a pre-defined number of months/years to the seller/financer.

After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Thus, it is of commercial interest to make the predictions.

Keywords-car; price; machine learning; artificial intelligence

seller/financers to be able to predict the salvage value (residual value) of cars with accuracy. If the residual value is under-estimated by the seller/financer at the beginning, the instalments will be higher for the clients who will certainly then opt for another seller/financer. If the residual value is over-estimated, the instalments will be lower for the clients but then the seller/financer may have much difficulty at selling these high-priced used cars at this over-estimated residual value. Thus, we can see that estimating the price of used cars is of very high commercial importance as well. Manufacturers from Germany made a loss of 1 billion Euros in their USA market because of mis-calculating the residual value of leased cars [3]. Most individuals in Mauritius who buy new cars are also very apprehensive about the resale value of their cars after certain number of years when they will possibly sell it in the used cars market.

Predicting the resale value of a car is not a simple task. It is trite knowledge that the value of used cars depends on a number of factors. The most important ones are usually the age of the car, its make (and model), the origin of the car (the original country of the manufacturer), its mileage (the number of kilometers it has run) and its horsepower. Due to rising fuel prices, fuel economy is also of prime importance. Unfortunately, in practice, most people do not know exactly how much fuel their car consumes for each km driven. Other factors such as the type of fuel it uses, the interior style, the braking system, acceleration, the volume of its cylinders (measured in cc), safety index, its size, number of doors, paint colour, weight of the car, consumer reviews, prestigious awards won by the car manufacturer, its physical state, whether it is a sports car, whether it has cruise control, whether it is automatic or manual transmission, whether it belonged to an individual or a company and other options such as air conditioner, sound system, power steering, cosmic wheels, GPS navigator all may influence the price as well. Some special factors which buyers

attach importance in Mauritius is the local of previous owners, whether the car had been involved in serious accidents and whether it is a lady-driven car. The look and feel of the car certainly contributes a lot to the price. As we can see, the price depends on a large number of factors. Unfortunately, information about all these factors are not always available and the buyer must make the decision to purchase at a certain price based on few factors only. In this work, we have considered only a small subset of the factors mentioned above. More details are provided in Section III.

This paper is organised as follows. In the next section, a review of related work is provided. Section III describes the methodology while in section IV, we describe, evaluate and compare different machine learning techniques to predict the price of used cars. Finally, we end the paper with a conclusion with some pointers towards future work.

2. Related Work

Surprisingly, work on estimated the price of used cars is very recent but also very sparse. In her MSc thesis [3], Listiani showed that the regression mode build using support vector machines (SVM) can estimate the residual price of leased cars with higher accuracy than simple multiple regression or multivariate regression. SVM is

better able to deal with very high dimensional data (number of features used to predict
the price) and can avoid both over-fitting and underfitting. In particular, she used a genetic algorithm to find the optimal parameters for SVM in less time. The only drawback of this study is that the improvement of SVM regression over simple regression was not expressed in simple measures like mean deviation or variance.

In another university thesis [4], Richardson working on the hypothesis that car manufacturers are more willing to produce vehicles which do not depreciate rapidly. In particular, by using a multiple regression analysis, he showed that hybrid cars (cars which use two different power sources to propel the car, i.e. they have both an internal
combustion engine and an electric motor) are more able to keep their value than traditional vehicles. This is likely due to more environmental concerns about the climate and because of its higher fuel efficiency. The importance of other factors like age, mileage, make and MPG (miles per gallon) were also considered in this study. He collected all his data from various websites.

Wu et al. [5] used neuro-fuzzy knowledge based system to predict the price of used cars. Only three factors namely: the make of the car, the year in which it was manufactured and the engine style were considered in this study. The proposed system produced similar results as compared to simple regression methods. Car dealers in USA sell hundreds of thousands of cars every year through leasing [6]. Most of these cars are returned at the end of the leasing period and must be resold. Selling these cars at the right price have major economic connotation for their

success. In response to this, the ODAV (Optimal Distribution of Auction Vehicles) system was developed by Du et al. [6]. This system not only estimates a best price for reselling the cars but also provides advice on where to sell the car. Since the United States is a huge country, the location where the car is sold also has a non-trivial impact on the selling price of used cars. A k-nearest neighbour regression model was used for forecasting the price. Since this system was started in 2003, more than two million vehicles have been distributed via this system [6].

Gonggi [7] proposed a new model based on artificial neural networks to forecast the residual value of private used cars. The main features used in this study were: mileage, manufacturer and estimate useful life. The model was optimised to handle nonlinear relationships which cannot be done with simple linear regression methods. It was found that this model was reasonably accurate in predicting the residual value of used cars.

3. Methodology

Data was collected from <<petites announces>> found in daily newspapers such as L'Express [8] and Le Defi [9]. We made sure that all the data was collected in less than one month interval as time itself could have an appreciable impact on the price of
cars. In Mauritius, seasonal patterns is not really a problem as this does not really affect the purchase or selling of cars. The following data was collected for each car: make, model, volume of cylinder (funnily this is usually considered same as horsepower in Mauritius), mileage in km, year of manufacture, paint colour, manual/automatic and price. Only cars which had their price listed were recorded.

Because many of the columns were sparse they were removed. Thus, paint colour and manual/automatic features were removed. The data was then further tweaked to
remove records in which either the age (year) or the cylinder volume was not available. Model was also removed as it would have been extremely difficult to get enough records for all the variety of car models that exist. Although data for mileage was sparse, it was kept as it is considered to be a key factor in determining the price of used cars

```
# Predicting used car prices

In this notebook, I'll work with the
[Kaggle](https://www.kaggle.com/avikasliwal/used-cars-price-prediction)
dataset about used cars and their prices. The notebook first includes
exploration of the dataset followed by prediction of prices.
## Import libraries
```

I'll import `datetime` to handle year, `numpy` to work with arrays and `pandas` to read in the dataset files, `matplotlib` & `seaborn` for plotting and `sklearn` for various machine learning models.

```
import datetime

import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score
```

Read dataset

I'll read the dataset and get information about it.

```
dataset = pd.read_csv("data/dataset.csv")
dataset.head(5)
```

Let's first split the dataset into train and test datasets.

```
X_train, X_test, y_train, y_test = train_test_split(dataset.iloc[:, :-1],
                                                    dataset.iloc[:, -1],
                                                    test_size = 0.3,
                                                    random_state = 42)
X_train.info()
```

Exploratory Data Analysis

Let's explore the various columns and draw information about how useful each column is. I'll also modify the test data based on training data.

Index

The first column is the index for each data point and hence we can simply remove it.

```
X_train = X_train.iloc[:, 1:]
X_test = X_test.iloc[:, 1:]
```

Name

Let's explore the various cars in the dataset.

```
X_train["Name"].value_counts()
```

As it appears, there are several cars in the dataset, some of them with a count higher than 1.
Sometimes the resale value of a car also depends on manufacturer of car and hence, I'll extract the manufacturer from this column and add it to the dataset.

```
make_train = X_train["Name"].str.split(" ", expand = True)
make_test = X_test["Name"].str.split(" ", expand = True)
X_train["Manufacturer"] = make_train[0]
X_test["Manufacturer"] = make_test[0]
```

Let's also confirm that there are no null values and identify all unique values.

```
plt.figure(figsize = (12, 8))
plot = sns.countplot(x = 'Manufacturer', data = X_train)
plt.xticks(rotation = 90)
for p in plot.patches:
    plot.annotate(p.get_height(),
                  (p.get_x() + p.get_width() / 2.0,
                   p.get_height()),
                  ha = 'center',
                  va = 'center',
                  xytext = (0, 5),
                  textcoords = 'offset points')

plt.title("Count of cars based on manufacturers")
plt.xlabel("Manufacturer")
plt.ylabel("Count of cars")
```

Maximum cars in the dataset are by the manufacturer **Maruti** and there are no null values.

I'll also drop the `Name` column.

```
X_train.drop("Name", axis = 1, inplace = True)
X_test.drop("Name", axis = 1, inplace = True)
```

Location

Location should not be a determinant for the price of a car and I'll safely remove it.

```
X_train.drop("Location", axis = 1, inplace = True)
X_test.drop("Location", axis = 1, inplace = True)
```

Year

Year has no significance on its own unless we try to extract how old a car is from this and see how its resale price may get affected.

```
curr_time = datetime.datetime.now()
X_train['Year'] = X_train['Year'].apply(lambda x : curr_time.year - x)
X_test['Year'] = X_test['Year'].apply(lambda x : curr_time.year - x)
```

Fuel_Type, Transmission, and Owner_Type

All these columns are categorical columns which should be converted to dummy variables before being used.
Kilometers_Driven

`Kilometers_Driven` is a numerical column with a certain range of values.
X_train["Kilometers_Driven"]
The data range is really varied and the high values might affect prediction, thus, it is really important that scaling be applied to this column for sure.
Mileage

This column defines the mileage of the car. We need to extract the numerical value out of each string and save it.
mileage_train = X_train["Mileage"].str.split(" ", expand = True)
mileage_test = X_test["Mileage"].str.split(" ", expand = True)

X_train["Mileage"] = pd.to_numeric(mileage_train[0], errors = 'coerce')
X_test["Mileage"] = pd.to_numeric(mileage_test[0], errors = 'coerce')
Let's check for missing values.
print(sum(X_train["Mileage"].isnull()))
print(sum(X_test["Mileage"].isnull()))
There is one missing value in each. I'll replace the missing value with the mean value of the column based on the training data.
X_train["Mileage"].fillna(X_train["Mileage"].astype("float64").mean(), inplace = True)
X_test["Mileage"].fillna(X_train["Mileage"].astype("float64").mean(), inplace = True)
Engine, Power and Seats

The `Engine` values are defined in CC so I need to remove `CC` from the data. Similarly, `Power` has bhp, so I'll remove `bhp` from it. Also, as there are missing values in `Engine`, `Power` and `Seats`, I'll again replace them with the mean.
cc_train = X_train["Engine"].str.split(" ", expand = True)
cc_test = X_test["Engine"].str.split(" ", expand = True)
X_train["Engine"] = pd.to_numeric(cc_train[0], errors = 'coerce')
X_test["Engine"] = pd.to_numeric(cc_test[0], errors = 'coerce')

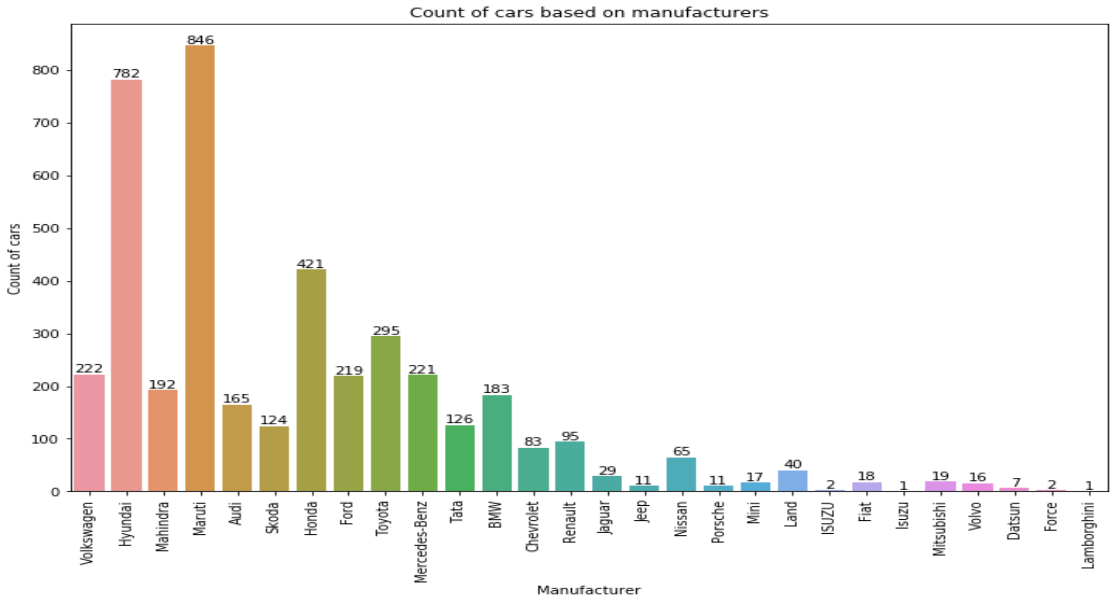
bhp_train = X_train["Power"].str.split(" ", expand = True)
bhp_test = X_test["Power"].str.split(" ", expand = True)
X_train["Power"] = pd.to_numeric(bhp_train[0], errors = 'coerce')
X_test["Power"] = pd.to_numeric(bhp_test[0], errors = 'coerce')

```
X_train["Engine"].fillna(X_train["Engine"].astype("float64").mean(), inplace
= True)
X_test["Engine"].fillna(X_train["Engine"].astype("float64").mean(), inplace =
True)

X_train["Power"].fillna(X_train["Power"].astype("float64").mean(), inplace =
True)
X_test["Power"].fillna(X_train["Power"].astype("float64").mean(), inplace =
True)

X_train["Seats"].fillna(X_train["Seats"].astype("float64").mean(), inplace =
True)
X_test["Seats"].fillna(X_train["Seats"].astype("float64").mean(), inplace =
True)
```
New Price

As most of the values are missing, I'll drop this column altogether.
```
X_train.drop(["New_Price"], axis = 1, inplace = True)
X_test.drop(["New_Price"], axis = 1, inplace = True)
```
Data Processing

Now that we have worked with the training data, let's create dummy columns for categorical columns before we begin training.
```
X_train = pd.get_dummies(X_train,
                         columns = ["Manufacturer", "Fuel_Type", "Transmission",
"Owner_Type"],
                         drop_first = True)
X_test = pd.get_dummies(X_test,
                         columns = ["Manufacturer", "Fuel_Type", "Transmission",
"Owner_Type"],
                         drop_first = True)
```
It might be possible that the dummy column creation would be different in test and train data, thus, I'd fill in all missing columns with zeros.
```
missing_cols = set(X_train.columns) - set(X_test.columns)
for col in missing_cols:
    X_test[col] = 0
X_test = X_test[X_train.columns]
```
Finally, as the last step of data processing, I'll scale the data.
```
standardScaler = StandardScaler()
standardScaler.fit(X_train)
X_train = standardScaler.transform(X_train)
X_test = standardScaler.transform(X_test)
```
Training and predicting

I'll create a **Linear Regression** model and a **Random Forest** model to train on the data and use it for future predictions.
linearRegression = LinearRegression()
linearRegression.fit(X_train, y_train)
y_pred = linearRegression.predict(X_test)
r2_score(y_test, y_pred)
rf = RandomForestRegressor(n_estimators = 100)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
r2_score(y_test, y_pred)
The **Random Forest** model performed the best with a R2 score of **0.88**.



GITHUBLINK:

https://github.com/IBM-EPBL/IBM-Project-19030-1659692233.git